Assignment 1

Aparna Anil Shelar(NU ID 001009233)

1. Problem

Calculate the mean distance between the lamp post and man who is taking series of the steps from lamp post of same length. that d is the Euclidean distance of the man from the lamp-post.

1. Task

In randomWalk() function loop over m elements which is a number of steps and called function randomMove(). randomMove is generation random moves i.e +1, -1 or 0 for each move. And calling move function which is incrementing/decrementing values of x and y as per the random value. Distance function calculating mean Euclidean distance of lamp post to person. Looped over 7 different m values and iterated for n is equal to 10.

Tabulated output and result has been showed with graph. Also ran unit tests which are all passing successfully.

1. Relationship Conclusion

d=f(n)

d=√n

1. Code

/\*

\* Copyright (c) 2017. Phasmid Software

\*/

package edu.neu.coe.info6205.randomwalk;

import java.util.Random;

public class RandomWalk {

private int x = 0;

private int y = 0;

private final Random random = new Random();

/\*\*

\* Private method to move the current position, that's to say the drunkard moves

\*

\* @param dx the distance he moves in the x direction

\* @param dy the distance he moves in the y direction

\*/

private void move(int dx, int dy) {

// TO BE IMPLEMENTED

x=x+dx;

y=y+dy;

//System.out.println(x);

//System.out.println(y);

}

/\*\*

\* Perform a random walk of m steps

\*

\* @param m the number of steps the drunkard takes

\*/

private void randomWalk(int m) {

for(int i=0;i<=m;i++)

{

randomMove();

}

}

/\*\*

\* Private method to generate a random move according to the rules of the situation.

\* That's to say, moves can be (+-1, 0) or (0, +-1).

\*/

private void randomMove() {

boolean ns = random.nextBoolean();

int step = random.nextBoolean() ? 1 : -1;

move(ns ? step : 0, ns ? 0 : step);

}

/\*\*

\* Method to compute the distance from the origin (the lamp-post where the drunkard starts) to his current position.

\*

\* @return the (Euclidean) distance from the origin to the current position.

\*/

public double distance() {

// TO BE IMPLEMENTED

double dis = Math.sqrt((y) \* (y) + (x) \* (x));

return dis;

}

/\*\*

\* Perform multiple random walk experiments, returning the mean distance.

\*

\* @param m the number of steps for each experiment

\* @param n the number of experiments to run

\* @return the mean distance

\*/

public static double randomWalkMulti(int m, int n) {

double totalDistance = 0;

for (int i = 0; i < n; i++) {

RandomWalk walk = new RandomWalk();

// walk.randomMove();

walk.randomWalk(m);

totalDistance = totalDistance + walk.distance();

}

return totalDistance / n;

}

public static void main(String[] args) {

// if (args.length == 0)

// throw new RuntimeException("Syntax: RandomWalk steps [experiments]");

//int m = Integer.parseInt(args[0]);

int arr[] = {100,200,50,70,20,10,80};

int n = 10;

if (args.length > 1) n = Integer.parseInt(args[1]);

for(int m = 0;m<=arr.length-1;m++)

{

//System.out.println(arr[m]);

double meanDistance = randomWalkMulti(arr[m], n);

System.out.println(arr[m] + " steps: " + meanDistance + " over " + n + " experiments");

}

}

}

1. Output

Text

Description automatically generated

1. Graph and Table

Graphical user interface, application, table, Excel

Description automatically generated

Chart, line chart

Description automatically generated

1. Test Cases:

Graphical user interface, text, application, email

Description automatically generated